International Journal of Innovative
Computing, Information and Control
Volume **9**, Number **3**, March **2013**

 ISSN 1349-4198
pp. **939–953**

BLIND BINARY WATERMARKING METHOD USING MAXIMUM WAVELET COEFFICIENT QUANTIZATION OF THE THIRD AND FOURTH SUB-BANDS

Payman Moallem[1], Ali Khodaie[2] and Ahmad Reza Naghsh-Nilchi[3]

[1]Department of Electrical Engineering
[3]Department of Computer Engineering
University of Isfahan
Isfahan, Iran
{ p_moallem; nilchi }@eng.ui.ac.ir

[2]Department of Electrical Engineering
Tiran Branch, Islamic Azad University
Isfahan, Iran
ali_khodaie2002@yahoo.com

Received October 2011; revised February 2012

Abstract. *To improve transparency, as an important parameter in watermarking, and maintain robustness, a new quantization coefficient of the third and fourth sub-bands of Discrete Wavelet Transform (DWT) is being proposed. In this method, all coefficients of four-level Haar DWT sub-bands of a host image (HL4, LH4, HL3 and LH3) are divided into different non-overlapping blocks. Then, each block is divided into some sets consisting of several wavelet coefficients as their members. Depending on whether a zero or one needs to be embedded, one or all sets are selected. By quantizing the first and second largest coefficient values in each selected set, a watermark bit is embedded. In decoding stage, the lowest difference between the first and second largest coefficient values in each block is compared with an empirical threshold, in order to estimate the watermark bit. In comparison with other methods, the implementation results show that the proposed method significantly improves transparency, while enhancing the robustness for most attacks. Moreover, the proposed method establishes a trade-off between transparency and robustness by tuning a threshold value in the decoding stage.*
Keywords: Blind binary watermarking, Haar wavelet transform, Sub-bands, Coefficients quantization

1. **Introduction.** In the current digital world, Internet piracy can easily modify an owner's documents without permission. This is a challenge met in every class of digital documents: whether text, audio, video, image or even the newly innovated 3D technology. In the recent years, watermarking has made possible a solution to protect the owners and producers of digital documents. In this trendy application of the copyright protection act, owner's identity, as a proof of ownership upon need, is embedded in the original document [1]. Today, digital images are the most common media used on the Internet. Hence, reserving the rights of their owners, through creating a strong watermarking design, is of high necessity. In the process of image watermarking, the original data is called the 'host image'; the embedded data is known as the 'watermark' and the output is labeled the 'watermarked image' [2].

Transparency and robustness are the most essential characteristics of a watermarking scheme. Transparency is the similarity of the watermarked image and the host image, while robustness means the ability of a retriever to extract the embedded watermark even

if the watermarked image is altered by some attacks [3]. Transparency is more important than robustness, since in low transparency, i.e., visible distortion of the watermarked image, the watermark can be detected and destroyed by an efficient attack [1]. Generally, transparency and robustness are in conflict, and as a result, establishing a trade-off between them can be considered as a state-of-the-art achievement in watermarking.

Watermarked images are constantly being attacked or modified. Overall, there are two attack categories; the first, known as non-malicious or non-hostile attack, is launched on the watermarked document by common practices such as JPG compression, geometric attacks, cropping, rotation and scaling [4,5]. The second category includes malicious attacks, which remove the watermark or make it unrecoverable. Malicious attacks, within themselves, are divided into two main groups: informed attacks and blind attacks. In an informed attack, the invader tries to extract the particular algorithm used for watermarking the asset, and then based on the exploited information it attempts to remove the watermark. On the other hand, in a blind attack, the goal of the attacker is to blindly remove the watermark [5].

Watermark schemes are proposed in two different domains: spatial and spectral-transform. As a fact, it is widely accepted that the robustness of methods based on spectral-transform excels the spatial ones [10,11]. Among those of spectral-transform, wavelet based algorithms are the most attractive, as in addition to having the general characteristics of Discrete Wavelet Transform (DWT), they are the most compatible with human visual system and compression standards [8,9].

In order to improve the robustness of DWT-based watermarking algorithms, the watermark is embedded in the most significant coefficients [10]. However, if the embedding stage selects global significant coefficients, finding those coefficients in the same order as the embedding stage after a blind attack, is not guaranteed [11]. To confront this challenge, choosing significant coefficients from separate blocks, is proposed [11]. In this approach, the order of watermark embedding is the same as the qualifying blocks. Furthermore, our simulations showed that extraction of each watermark bit is independent of others, meaning a wrong extraction does not affect the right extraction of others. The proposed method in [11] does not have effective robustness against most non-hostile attacks. Moreover, its transparency is not high enough to withdraw distortions.

In this article, so as to improve the robustness, the third and fourth sub-bands are quantized simultaneously. In order to reach acceptable transparency, two different inserting algorithms are devised, where the low complexity algorithm is assigned to embed the majority of the bits. Considering the limitations on these two embedding algorithms, some unnecessary changes are prevented, hence, improving transparency. In this method, the significant coefficients are chosen from the separate HL4, LH4, HL3, and LH3 block sub-bands. Subsequently, each watermark bit is embedded in its respective block using different inserting algorithms for the majority and minority bits. A trade-off point between robustness and transparency can be adjusted by a parameter termed PCM. In comparison with earlier methods, the herein proposed method shows higher significance in fidelity and more robustness against most regular watermarking attacks.

In this article, Section 2 gives an in-depth description of the embedding and extracting processes of the proposed method. Its performance is later evaluated in Section 3; along with some simulation results with respect to fidelity and robustness for different binary watermarks and host images and finally, there is a section on conclusion.

2. **The Proposed Method.**

The overall plan of the proposed method has four stages in its embedding process (shown in Figure 1(a)). The first stage transforms the host image into corresponding Haar wavelet coefficients. In the second stage, blocks and sets are made

from HL4, LH4, HL3, LH3 sub-bands. A comprehensive description of the above stages is provided in the 'set making' subsection. The next stage utilizes two different algorithms to insert the binary watermark. The first and second algorithms are designed to insert the majority and minority of the watermark bits, respectively. Therefore, summation of changes in coefficients is lessened which consequently improves the transparency of the watermarked image. In the final stage, the watermarked image is obtained by inverse DWT of all wavelet sub-bands. An all-inclusive report of the third and fourth stages is presented in the 'embedding' subsection.

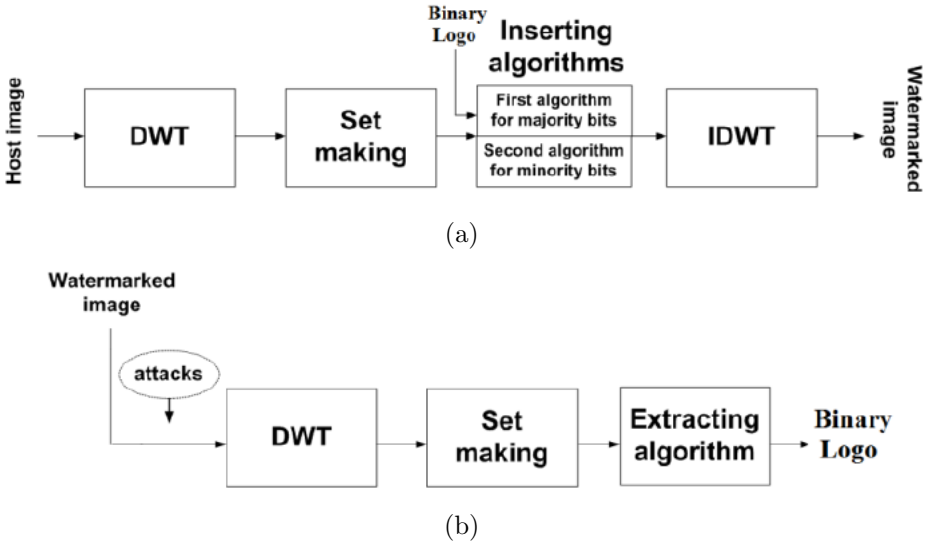

FIGURE 1. The proposed watermark method, (a) embedding process and (b) extraction process

Figure 1(b) illustrates the extraction process of the watermark from a watermarked image, possibly under attack or modification. Identical to the embedding process, DWT and set making are the first two stages of the extraction process. The 'extraction process' subsection, fully explains the finishing stage of this process, associated with extracting the algorithm.

2.1. **Set making.** Prior to the set making stage, the host image is transformed into corresponding wavelet coefficients using the four-level Haar DWT. Then, the HL4, LH4, HL3, LH3 coefficient sub-bands are reshaped into four sequences of coefficients. Concatenating the four sequences composes a super block (SB). For a watermark through NW binary bits, the super block is divided into NW blocks and each bit of watermark is inserted in one block. Available Coefficients (AC) in a super block are computed by Equation (1),

$$AC = AC4 + AC3 \tag{1}$$

where AC4 stands for all available coefficients of HL4 and LH4 sub-bands and AC3 represents every available coefficient of HL3 and LH3 sub-bands. AC4 and AC3 are obtained using Equations (2) and (3), respectively.

$$AC4 = 2 \times \frac{I_W \times I_L}{2^4 \times 2^4} \tag{2}$$

$$AC3 = 2 \times \frac{I_W \times I_L}{2^3 \times 2^3} \tag{3}$$

In these equations, I_W and I_L correspond to the width and the length of the host image. Equation (4), calculates the number of necessary blocks.

$$NW = W_W \times W_L \tag{4}$$

W_W and W_L symbolize the width and length of the watermark, respectively. Considering the available coefficients (AC) and the number of blocks (NW), determining the number of coefficients per block (NPB) is possible via Equation (5).

$$NPB = \frac{AC}{NW} \tag{5}$$

At this point, each block is divided into S sets of M members, while S is selected to be narrowly less than M. For instance, in order to embed a 16×32 binary watermark in a 512×512 host image, there are 10240 available coefficients (AC = AC3+AC4 = $2 \times 4096 + 2 \times 1024 = 10240$), whereas the number of necessary blocks is 512 (NW = $16 \times 32 = 512$). In this case, the NPB is 20 (NPB = AC/NW = 10240/512) and therefore, this block is divided into 4 sets, each inclusive of 5 members. Figure 2 shows the set making process of a sample 512×512 host image and a 16×32 watermark. Figure 3 demonstrates the reshaping process of a sample watermark and how each bit of the watermark is assigned to its corresponding block.

2.2. **Inserting algorithm.** The main objective of an inserting algorithm is to introduce a bit into each set-contained block. Initially, the first, second and third local maximums of each set are found. The difference between first and second local maximums of each

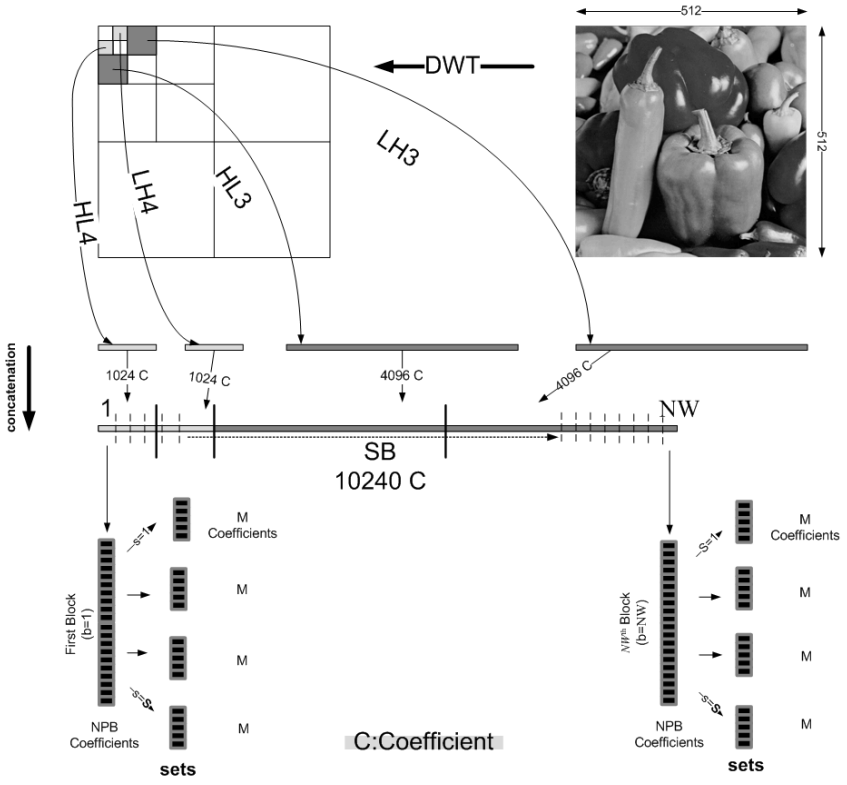

FIGURE 2. Set making for a sample 512×512 host image and a 16×32 watermark image

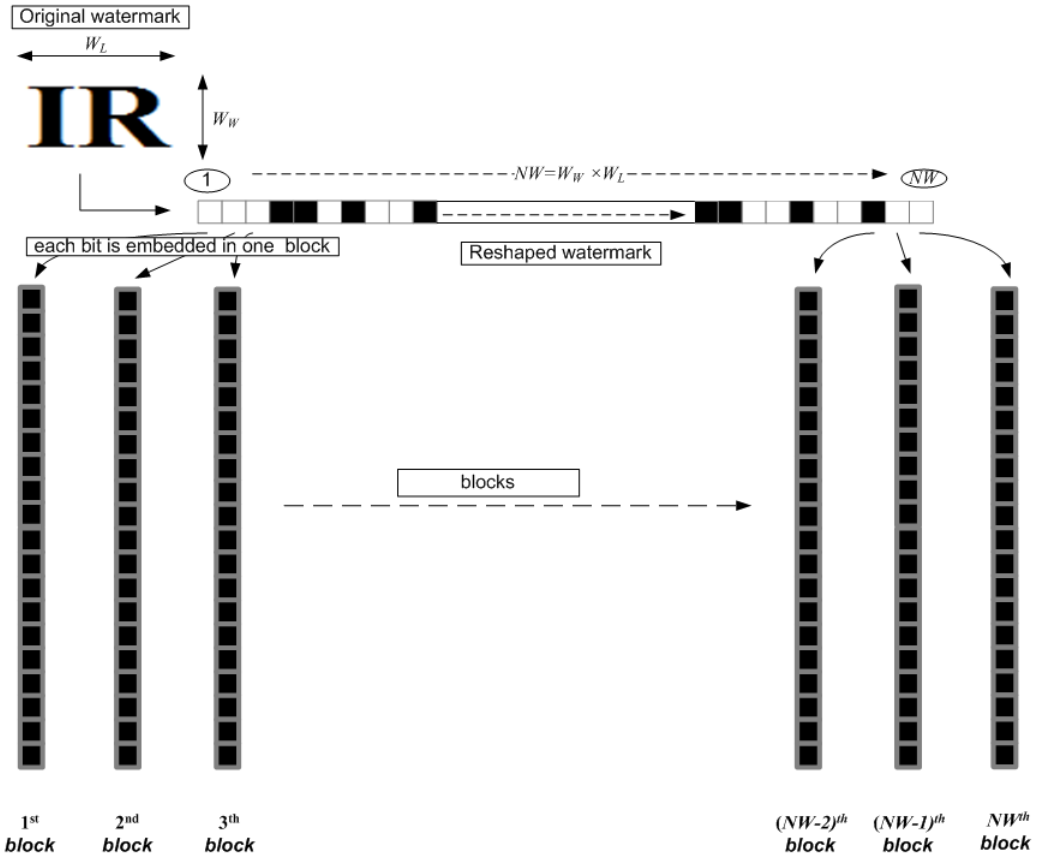

FIGURE 3. Reshaping process of a sample watermark image and assigning each bit of the watermark to its equivalent block

set, also known as the difference of maximums, is gained by Equation (6).

$$dif(b,s) = max(b,s) - smax(b,s) \tag{6}$$

where $max(b,s)$ and $smax(b,s)$ are the first and second local maximums of the sth set in the bth block, respectively.

The watermark is reshaped into a sequence of bits; then, through re-quantizing the first and second local maximums in each block, these bits are embedded. In a binary watermark, there are only two types of information, 1's and 0's; thus, with the intention of increasing the robustness, two different inserting algorithms with opposing behaviors are proposed. The first algorithm is applied in order to decrease $dif(b,s)$, by re-quantization of $max(b,s)$ and $smax(b,s)$. On the contrary, the second one is used to increase $dif(b,s)$, through much the same process. Meanwhile, the permitted change in local maximums (PCM) is a factor for establishing a trade-off between transparency and robustness. For some sets, $max(b,s)$ and $smax(b,s)$ of the second inserting algorithm, should be modified to $max(b,s) + \text{PCM}$ and $smax(b,s) - \text{PCM}$, respectively. The PCM value is dependent on how much transparency or robustness is needed. To facilitate an acceptable transparency (i.e., less modification) considering a constraint on the PCM is a must.

2.2.1. *The first inserting algorithm.* For the bth block, Equation (7) calculates $mindf(b)$, the minimum value of $dif(b,s)$ over s.

$$mindf(b) = \min_{s=1}^{S} \{dif(b,s)\} \tag{7}$$

Obtained by Equation (8), $s_{md}(b)$ represents a set where $dif(b, s_{md}(b))$ is equal to $mindf(b)$.

$$s_{md}(b) = \arg\left(\min_{s=1}^{S}\{dif(b,s)\}\right) \tag{8}$$

In each block, the first inserting algorithm only modifies $dif(b, s_{md}(b))$ for the set with the minimum difference between its first and second maximums. To decrease $dif(b, s_{md}(b))$ in the first algorithm, $max(b, s_{md}(b))$ is decreased while $smax(b, s_{md}(b))$ is increased as presented by Equation (9),

$$\begin{aligned} maxn(b, s_{md}) &= max(b, s_{md}) - \text{PCM} \\ smaxn(b, s_{md}) &= smax(b, s_{md}) + \text{PCM} \end{aligned} \tag{9}$$

where $maxn(b, s_{md}(b))$ and $smaxn(b, s_{md}(b))$ are the first and second maximums of the (s_{md})th set of the bth block after re-quantizing. Equation (10) is used, once $mindf(b)$ is less than $2\times$PCM. In this scenario, the new values of the first and second local maximums are equal, and the new value of $mindf(b)$ is zero.

$$maxn(b, s_{md}) = smaxn(b, s_{md}) = \frac{max(b, s_{md}) + smax(b, s_{md})}{2} \tag{10}$$

It is noteworthy that in the first inserting algorithm, shown in Figure 4, only two coefficients are changed to embed a bit. Figure 5 shows an example of embedding one bit in the bth block of a super block, containing 4 sets of 5 coefficients each. Here, the first and second maximums of the 4th set must be quantized, since this set contains the minimum $dif(b, g)$ of the block. Also, as $dif(b, 4) < 2 \times$ PCM, Equations (10) is used. Quantizing through the first inserting algorithm and applying it to the majority of bits, achieves excellent transparency.

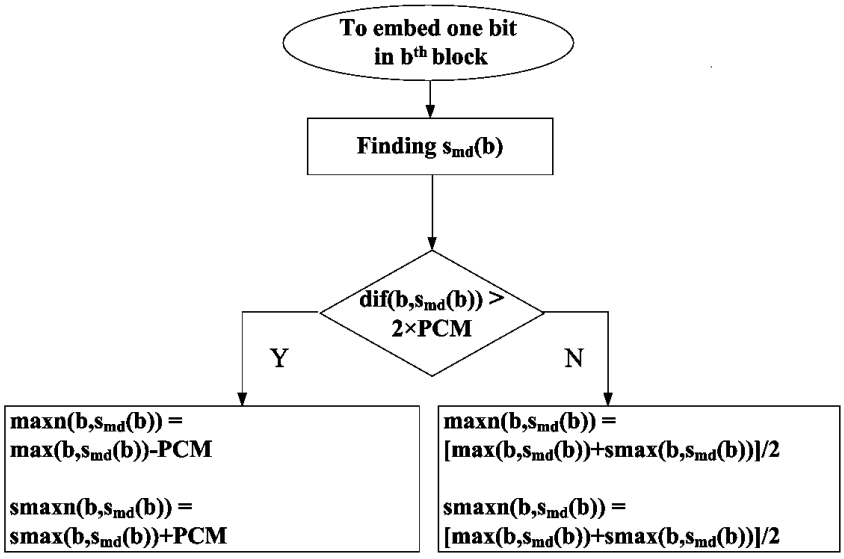

FIGURE 4. The first inserting algorithm for embedding a bit in the bth block

2.2.2. *The second inserting algorithm.* To apply the second inserting algorithm for embedding a block, it is assumed that the difference between the first and second local maximums of each set is appropriately high. In this case, it is not necessary to make any changes in $max(b, s)$ and $smax(b, s)$. In regards to the PCM value, when $dif(b, s)$ is greater than 2PCM, $max(b, s)$ and $smax(b, s)$ need no change during the embedding procedure. On the other hand, where $dif(b, s)$ is smaller than 2PCM, it needs an increase,

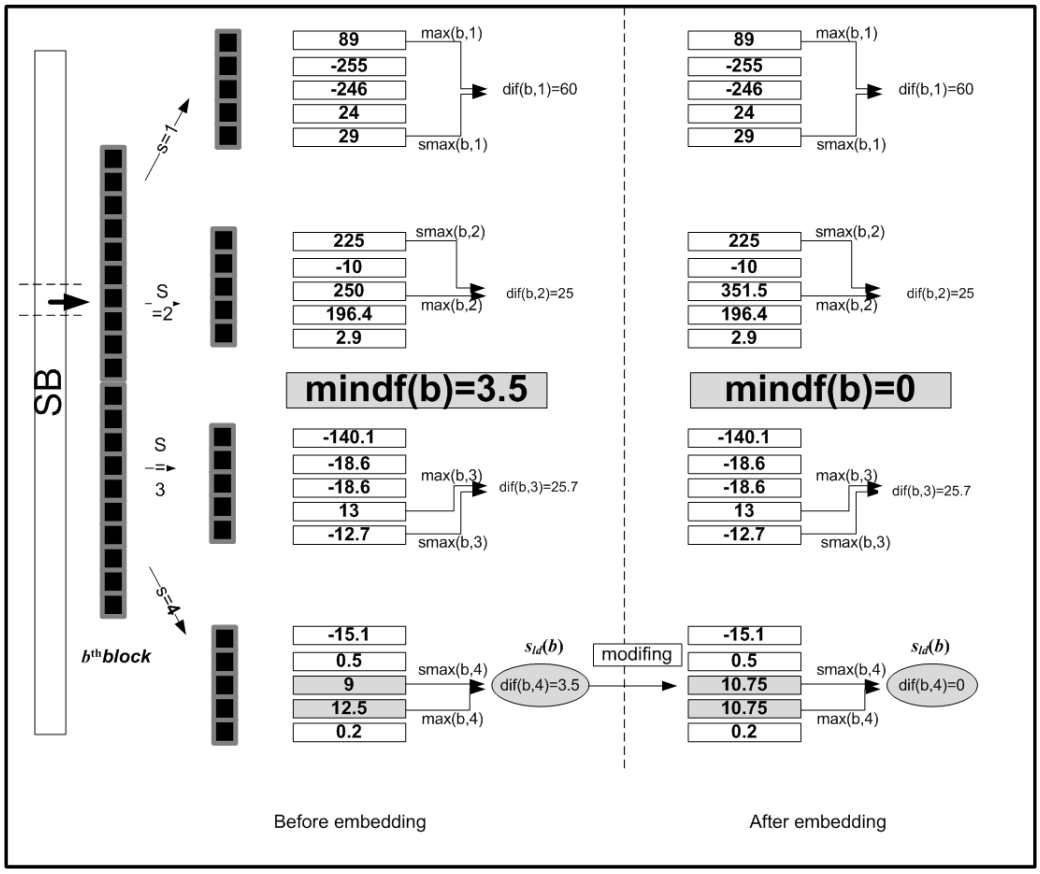

FIGURE 5. An example for embedding a bit in the bth block through the first inserting algorithm

which means $max(b, s)$ must be increased to reach $max(b, s) + \text{PCM}$, and $smax(b, s)$ needs to decrease by the value of PCM. A decrease in $smax(b, s)$ causes a problem after re-quantization, which needs assuring that the new value of $smax(b, s)$ remains the second maximum of the block. Therefore, it is also necessary to check the third local maximum ($thmax(b, s)$) in the second inserting algorithm. The difference between the second and third maximums of each set in a block is obtained by Equation (11).

$$dif2(b, s) = smax(b, s) - thmax(b, s) \tag{11}$$

If the corresponding $dif(b, s)$ of a set is less than $2 \times \text{PCM}$, and its $dif2(b, s)$ is greater than PCM, Equation (12) is used to adjust the first and second local maximums in order to embed a bit in the block using the second algorithm (shown in Figure 6).

$$\begin{aligned} maxn(b, s) &= max(b, s) + \text{PCM} \\ smaxn(b, s) &= smax(b, s) - \text{PCM} \end{aligned} \tag{12}$$

Equations (13) is used once $dif(b, s)$ is less than $2 \times \text{PCM}$ and $dif2(b, s)$ is less than PCM.

$$\begin{aligned} maxn(b, s) &= max(b, s) + \text{PCM} \\ smaxn(b, s) &= thmax(b, s) \end{aligned} \tag{13}$$

No change is necessary for all other sets.

So, in sets where $dif(b, s)$ is less than $2 \times \text{PCM}$, all $max(b, s)$ need to change. On the other hand, no change is required when $dif(b, s)$ is greater than $2 \times \text{PCM}$.

Figure 7 shows the second algorithm for embedding one bit in the bth block of a super block containing 4 sets of 5 coefficients (shown in Figure 5). In the first set of this example

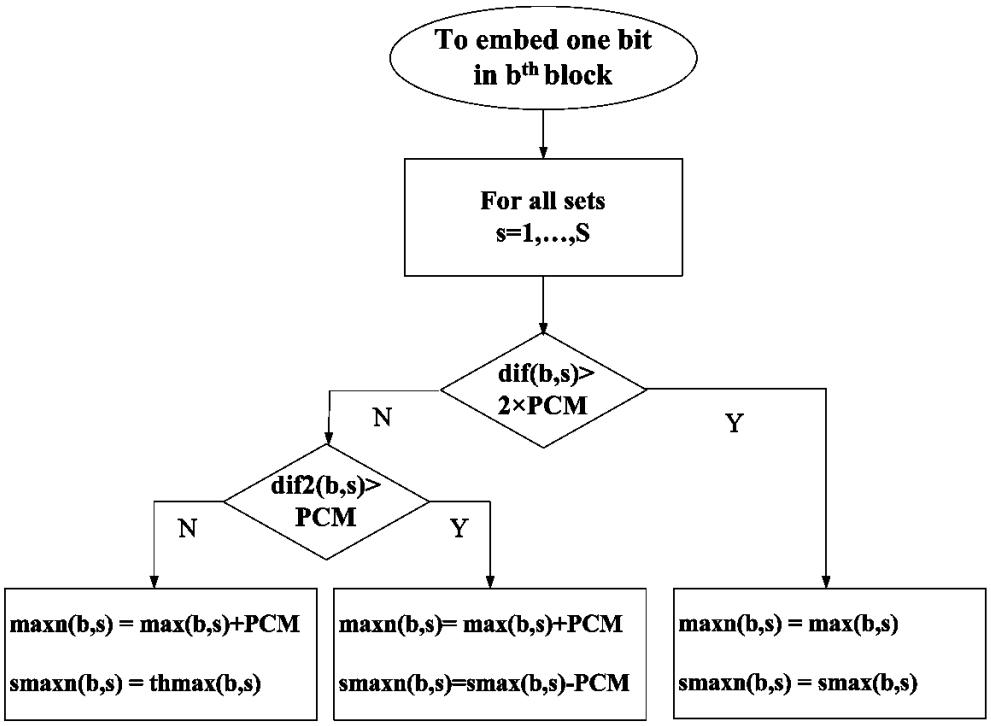

FIGURE 6. The second algorithm of embedding a bit in the bth block

$dif(b,1) > 2 \times$ PCM, meaning no change is necessary, while for other sets $dif(b,s) < 2 \times$ PCM, where quantizing the first and second maximums are necessary. In the second set, difference between the second and third local maximums ($dif2(b,2)$) is more than PCM.

In this case, the third maximum does not limit quantizing the second maximum. Therefore, Equation (12) is used. In sets 3 and 4, $dif2(b,s) <$ PCM, making Equation (13) functional. In these sample blocks, 3 changes were made in the sets, two of which were limited since the third maximum affected the quantization of the second maximum.

In effect, taking the second and third maximums into consideration of the proposed inserting algorithm limits the modifications. Consequently, these strategies cause significant transparency improvements. There are, though in some cases, blocks which require no change over their sets.

2.2.3. *The inserting algorithms selection.* The important question on inserting algorithms is 'which of them should be selected for 1 and which for 0 in a binary watermark?' As mentioned before, transparency is of high importance in watermarking. In most cases, minor transform changes could result in higher transparency of a watermarking algorithm. Since it only makes changes in one set of each block, the transparency of the first inserting algorithm is higher than that of the second. The second inserting algorithm requires a difference between the first and second maximums in all sets of each block. Therefore, to have a higher transparency in the proposed embedding process, the first and second algorithms are respectively specified for the majority and minority of the watermark bits. Consequently, this method starts with analyzing the number of 0's and 1's in the binary watermark. This study supposes that the number of 0's is greater than 1's for bench mark logos; however, for other logos, this assumption may not be true. So, in bench mark logos, the majority of the bits are 0, inserted by the first inserting algorithm.

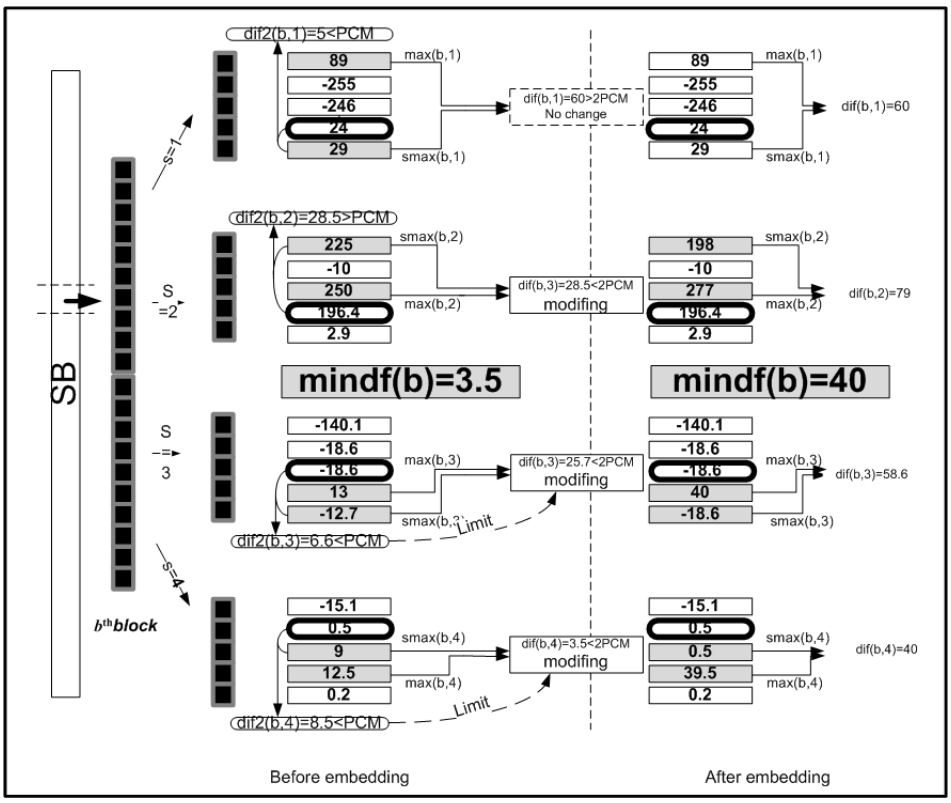

FIGURE 7. An example of embedding a bit in the bth block using the second inserting algorithm, PCM is 27.

2.3. **Extraction process.** Initially, the extraction process uses the four levels of Haar DWT to transform the watermarked image into wavelet coefficients; then, a superblock is made and blocks and sets are generated in the same manner as the set making stage of the embedding process. For every block, $mindf(b)$ is computed. If it is less than the detection threshold (TR), the embedded bit is estimated 0, otherwise 1 (for majority of a bench mark watermark bits, it is 0). The complete extraction process for the bth block is shown in the following code;

if $(mindf(b) < TR)$ then $WB's(b) = 0$
else $WB's(b) = 1$

In this code, $WB's(b)$ and $mindf(b)$ correspond to the bth bit of extracted watermark sequence and minimum of $dif(b, s)$ over s, respectively. The TR should correlate to the predefined PCM. In this article, TR is empirically obtained by $TR = 0.4$ PCM. Finally, the extracted watermark sequence is reshaped to its origin.

3. **Simulations and Results.** All simulations of the embedding, attacking and extraction processes are performed in Matlab R2007b environment. In order to evaluate the robustness, median filters, JPEG compression, sharpening, Gaussian filter, geometric attacks, average filter, wiener filter and three hybrid attacks are applied. 1/4 of the host image from the upper left corner is cropped and replaced by zeros, to imitate a cropping attack. For a scaling attack, the host image is scaled down to 256×256 and then rescaled to its original 512×512 size. In the simulation of a rotation attack, the host image is rotated around its upper left corner.

The comparison of different watermarking methods, took place on three standard 512×512 images, namely Lena, Peppers and Barbara. Figure 8 illustrates the three different

binary logos used as watermarks in this study. As shown, 0's constitute the majority of the bits.

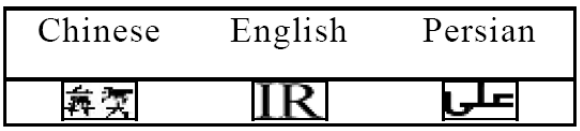

FIGURE 8. Shape of Chinese, English and Persian binary watermarks

Normalized Cross Correlation (NCC) evaluates a method's robustness through Equation (14),

$$NCC = \frac{\sum_{i=1}^{W_W} \sum_{j=1}^{W_L} \left[WB'(i,j) \times WB(i,j)\right]}{\sum_{i=1}^{W_W} \sum_{j=1}^{W_L} \left[WB(i,j)\right]^2} \tag{14}$$

where $WB(i,j)$ and $WB'(i,j)$ are, correspondingly, the (i,j)th pixels of the original and extracted logo. The fidelity measurement of a watermarked image is obtained by *PSNR* function, represented in Equation (15),

$$PSNR(I,I') = 10\log_{10}\left(\frac{255^2}{MSE}\right) \tag{15}$$

where I and I' are the host image and watermarked image, respectively. *MSE* is obtained by Equation (16),

$$MSE = \frac{1}{I_W \times I_L} \sum_{i=1}^{I_W} \sum_{j=1}^{I_L} \left(I(i,j) - I'(i,j)\right)^2 \tag{16}$$

where $I(i,j)$ and $I'(i,j)$ are, in that order, the (i,j)th pixels of the host image and watermarked image.

In the proposed watermarking method, an increase in PCM causes a robustness boost, as well as a declined fidelity, however. In order to have significant and simultaneous robustness and fidelity, PCM is empirically set to 27 for 512×512 8-bit, gray scale, testing host images.

The first part of the simulation includes a Chinese 16×32 logo, while the second part includes an English and a Persian logo as binary watermarks. In the third part, the size of the Chinese logo is enlarged to 32×32 and then, the number of its majority bits is increased. Lastly, as a control parameter for establishing a trade-off between transparency and robustness, the PCM effect variation is analyzed.

To improve the fidelity of a watermarked image, it is best to use the first inserting algorithm, since the number of its modified wavelet coefficients is less than the second algorithm. In this article, mMR is defined as the ratio of minority to majority watermark bits, stating that mMR and transparency are inversely proportional.

3.1. **Embedding the 16×32 Chinese watermark.** This simulation presents a comparison between the herein proposed method and three other blind wavelet based methods for embedding a 16×32 Chinese binary watermark in a 512×512 Lena host image. Two out of those three methods (described in [1,12]), similar to the proposed method, use the four level DWT, while the other adopts a block based algorithm to embed the watermark [11]. In [12] the wavelet coefficients are divided into some super trees, each of which

includes coefficients of the second, third and fourth level of DWT. Then by making statistical difference between each pair of super trees, one bit is embedded. Another method quantizes the distance between the two smallest coefficients of a tree in order to embed a watermark bit in that tree [1]. In [11], random selection of various sized blocks from HL3 or LH3 sub-bands, followed by the quantization of difference between the first and second maximums in each block, embeds a watermark bit.

To draw a fair comparison, in this simulation, the Chinese watermark is modified manually so that the corresponding mMR reaches 1.

This allows increasing the PCM in order to have better robustness, while maintaining an acceptable fidelity. In fact, worst case scenario in the proposed method, is the equal number of majority and minority bits (i.e., mMR = 1). In [1], the mMR is reported 1, whereas in [11] this ratio is not reported at all. In order to model this worst case, a watermark similar to the one used in [11] is chosen and then adjusted to match the required mMR.

As previously explained, the host is a 512×512 image, and since there are 512 bits in the watermark image, the NPB is 20. Therefore, each block is divided into 4 sets of 5 wavelet coefficients.

Table 1 presents the transparency (PSNR by Equation (15)) and robustness (NCC by Equation (14)) comparison of the methods. As far as transparency is concerned, the herein method shows superior and higher performance than others. Moreover, this method presents a higher robustness for all kinds of attacks than the method in [12]. In judgment with the methods described in [1,11], this proposed method shows a higher robustness against most attacks.

Figure 9 shows the robustness (measured by NCC) of the proposed method opposing five other attacks for the Lena, Peppers and Barbara 512×512 host images. The fidelity (measured by PSNR) of the watermarked images are 47.29db, 47.15db and 46.82db, respectively, representing the sufficient robustness of the proposed method for all attacks.

3.2. **Embedding the 16×32 English and Persian watermarks.** The watermarking process is repeated using two other 16×32 watermarks shown in Figure 8. These two

TABLE 1. Comparing the proposed method with similar methods for the 512×512 Lena host image and the Chinese watermark 16×32 (NA mean not accessible)

Method / **Attacks**	**[16] PSNR = 38.20**	**[4] PSNR = 44.73**	**[15] PSNR = 42.02**	**Proposed method PSNR = 47.29**
Median filter (3×3)	0.51	0.92	0.9	0.94
Median filter (4×4)	0.23	0.75	0.76	0.77
Median filter (7×7)	NA	NA	0.53	0.35
JPEG (QF = 10)	NA	0.33	0.34	0.40
JPEG (QF = 20)	NA	0.59	0.67	0.81
JPEG (QF = 30)	0.15	0.81	0.82	0.83
JPEG (QF = 50)	0.26	0.95	0.96	0.97
JPEG (QF = 70)	0.57	1	0.97	0.98
JPEG (QF = 90)	1	1	0.99	0.98
Sharpening	0.46	0.99	0.97	0.84
Gaussian filter	0.64	0.96	0.88	0.95
Rotation (degree: +0.25)	0.37	0.61	0.59	0.79
Rotation (degree: −0.25)	0.32	0.65	0.6	0.78
Cropping 1/4	NA	0.60	0.66	0.62
Scaling 256×256	NA	0.86	0.88	0.96

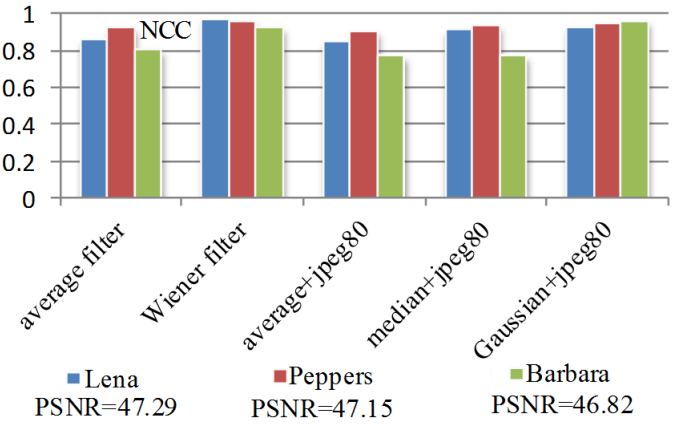

FIGURE 9. Simulation results for the Lena, Peppers and Barbara host image versus 5 attacks

watermarks include 150 zeros out of a total of 512 binary bits, meaning that the corresponding mMR is about 0.41. For visual comparison, Table 2 illustrates the extracted English watermark after different attacks.

The results of embedding a 150-zero Chinese watermark (mMR = 0.41) is compared with the English and Persian ones in Figure 10. Based on the results, the performance of the proposed method is almost independent of the order of 1's and 0's in a watermark shape.

3.3. **Embedding a larger sized Chinese watermark.** To evaluate the capability of the proposed method, size of the Chinese watermark is enlarged to 32×32, i.e., the number of coefficients in each block is reduced to 10. Each block is divided into 2 sets of 5 coefficients. Figure 11 illustrates the robustness and transparency of this method once embedding different sizes of the Chinese watermark on the Lena host image. Compared to the smaller watermark, the fidelity of larger one is decreased by about 1.2db.

Clearly, there is an upper limit to the number of watermark bits. Theoretically, there should be at least two sets, each containing three coefficients. Therefore, the maximum number of watermark bits (NWmax) can be calculated through $\text{NWmax} = (\text{AC})/(2 \times 3)$; hence, it is impossible to make sets if the number of watermark bits is greater than NWmax.

TABLE 2. The extracted English watermark after different attacks (mMR = 0.41)

Attack	Watermark	Attack	Watermark
Median filter (3 × 3)		Gaussian filter	
Median filter (4 × 4)		Rotation (degree:+0.25)	
Median filter (7 × 7)		Rotation (degree:-0.25)	
JPEG 10		Cropping ¼	
JPEG 20		Scaling 256 × 256	
JPEG 30		average filter	
JPEG 50		wiener filter	
JPEG 70		average+jpeg80	
JPEG 90		median+jpeg80	
Sharpening		gaussian+jpeg80	

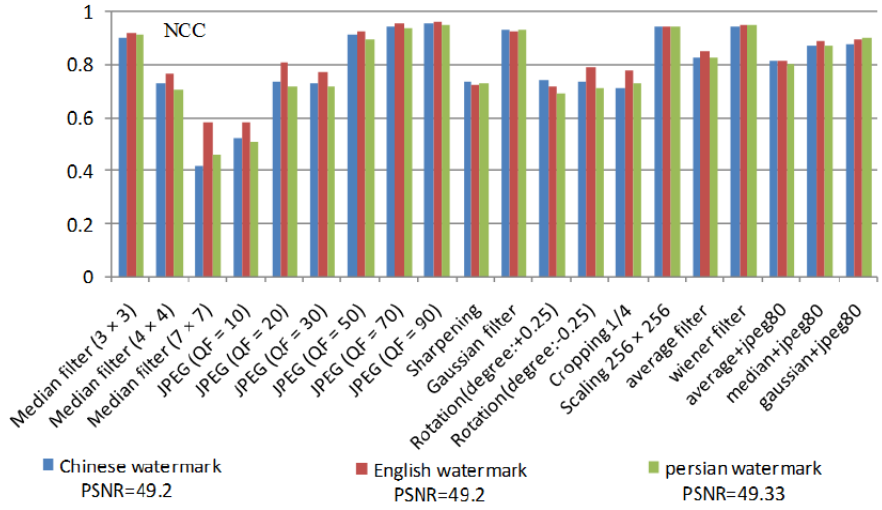

FIGURE 10. Comparing the effects of the watermark shape on robustness and fidelity

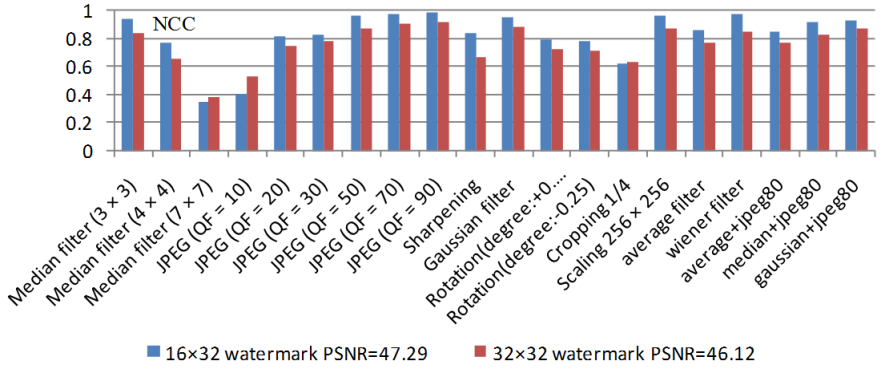

FIGURE 11. Comparing the effects of watermark size on robustness and fidelity

3.4. **Analyzing the effects of the mMR value.** As mentioned before, mMR signifies the number of minority bits of a watermark, which has a strong effect on the robustness and transparency of the proposed method. Figure 12 analyzes the effects of decreasing the mMR of the Chinese watermark (thus increasing the majority bits) on robustness and fidelity. As predicted, a decreased mMR means the fidelity is increased, since the first algorithm is used more often. All simulations had a PCM value of 27.

3.5. **Analyzing the effects of the PCM value.** The permitted change on maximums, PCM, is a parameter used for establishing a trade-off between transparency and robustness. This simulation studies the effects of changes in the PCM value for embedding the Chinese watermark with an mMR of equal to 1, on the Lena host image. The PCM value is directly proportional to robustness and inversely proportional to fidelity. Figure 13 shows the results.

4. **Conclusion.** This article proposes a new blind block-based watermarking method using the four levels of DWT, for copyright protection applications. In the embedding process, a super block from HL4, LH4, HL3 and LH3 is generated and is then divided into some set-containing blocks. In order to improve the robustness of this method over similar methods, quantization of the fourth sub-band coefficients is also used. Also, the first and second local maximums of some sets in each block are modified. Each

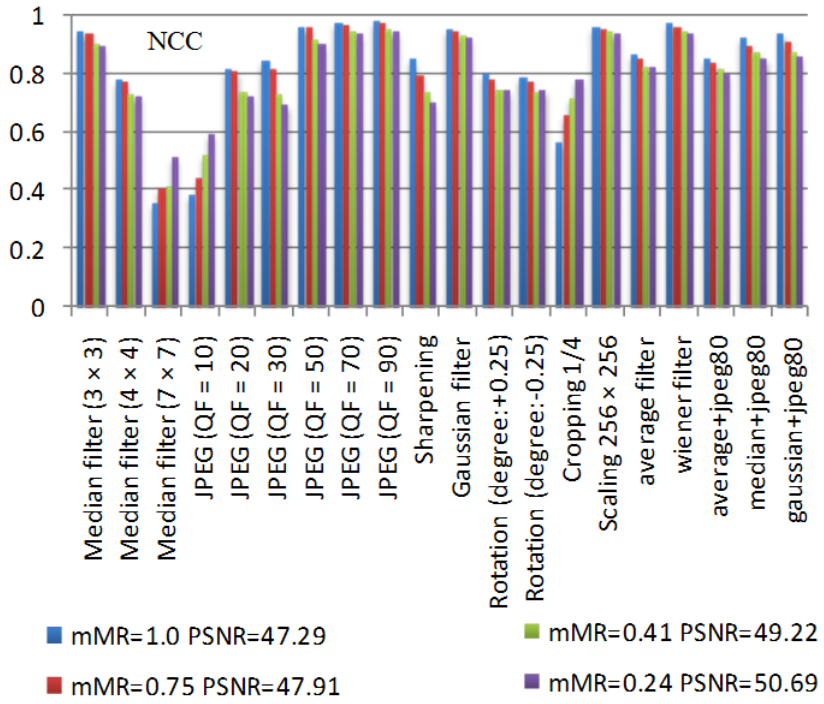

FIGURE 12. Analyzing the effects of mMR on fidelity and robustness

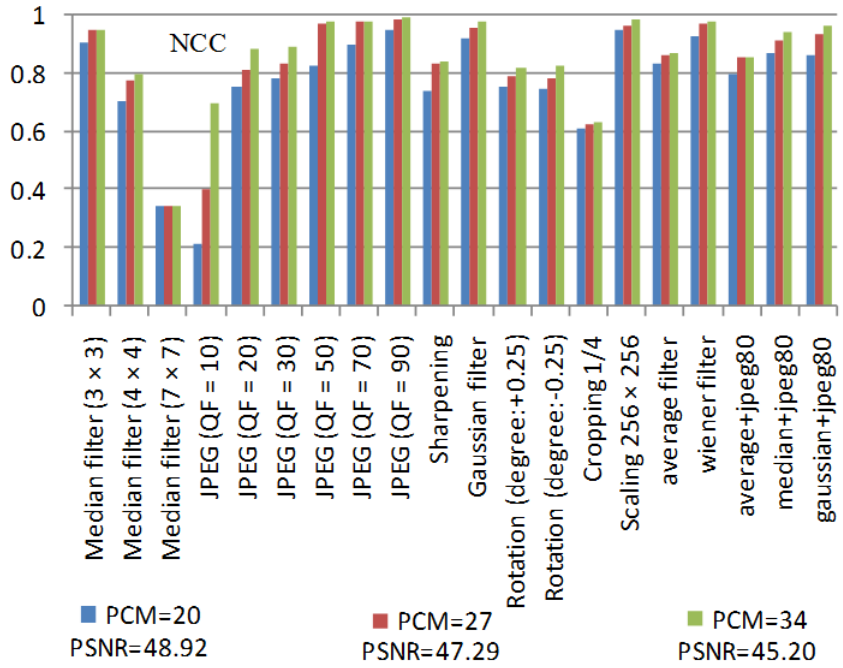

FIGURE 13. Analyzing the effects of the PCM value on fidelity and robustness

watermark bit can be embedded in a block by means of two inserting algorithms. The first inserting algorithm is used to embed the majority of watermark bits, since it creates less modification. In fact, both of these algorithms could be used to embed 0's and 1's adaptively, depending on whether 0's are the majority or 1's. Furthermore, in the second algorithm, some unnecessary changes were prohibited by considering the difference between the first and second and also the second and third maximums. Of the two proposed algorithms, applying the least complex one to the majority of bits, improves

transparency. In order to establish a trade-off between transparency and robustness, the permitted change in local maximums is defined as a control parameter, and used during the embedding process. After re-quantizing the first and the second local maximums of some sets, the watermarked image is achieved through inverse DWT.

Finally, the proposed method is judged against some other DWT-based methods regarding the embedment of various binary watermark images with different sizes and various host images. Simulation and comparison results indicate that the proposed method significantly improves fidelity while proving higher robustness against most tested attacks.

REFERENCES

[1] W.-H. Lin, Y.-R. Wang and S.-J. Horng, A wavelet-tree based watermarking method using distance vector of binary cluster, *Expert Systems with Applications*, vol.36, pp.9869-9878, 2009.

[2] Y. Yan, W. Cao and S. Li, Block-based adaptive image watermarking scheme using just noticeable difference, *Proc. of International Workshop on Imaging Systems and Techniques*, Shenzhen, China, pp.377-380, 2009.

[3] C. Chemak, M. SalimBouhlel and J. C. Lapayre, A new scheme of robust image watermarking: The double watermarking algorithm, *Proc. of Summer Computer Simulation Conference*, San Diego, California, pp.1201-1208, 2007.

[4] G. Doërr and J.-L. Dugelay, A guide tour of video watermarking, *Signal Processing: Image Communication*, vol.18, no.4, pp.263-282, 2003.

[5] M. Barni and F. Bartolini, *Watermarking Systems Engineering: Enabling Digital Assets Security and Other Applications*, CRC Press, New York, 2004.

[6] G. Bhatnagar and B. Raman, A new robust reference watermarking scheme based on DWT-SVD, *Computer Standards and Interfaces*, vol.31, pp.1002-1013, 2009.

[7] L. Li, H.-H. Xua, C.-C. Changb and Y.-Y. Ma, A novel image watermarking in redistributed invariant wavelet domain, *Journal of Systems and Software*, vol.84, pp.923-929, 2011.

[8] H.-F. Li, N. Chang and X.-M. Chen, A study on image digital watermarking based on wavelet transform, *Journal of China Universities of Posts and Telecommunications*, vol.17, pp.122-126, 2010.

[9] J.-J. Lee, N.-Y. Lee, W. Kim and G.-Y. Kim, A new incremental watermarking based on dual-tree complex wavelet transform, *Journal of Supercomputing*, vol.33, pp.133-140, 2005.

[10] E. Ganic and A. M. Eskicioglu, Robust DWT-SVD domain image watermarking: Embedding data in all frequencies, *ACM Multimedia and Security Workshop*, Magdeburg, Germany, 2004.

[11] W.-H. Lin, Y.-R. Wang, S.-J. Horng, T.-W. Kao and Y. Pan, A blind watermarking method using maximum wavelet coefficient quantization, *Expert Systems with Applications*, vol.36, pp.11509-11516, 2009.

[12] S. H. Wang and Y. P. Lin, Wavelet tree quantization for copyright protection watermarking, *IEEE Trans. on Image Processing*, vol.13, no.2, pp.154-165, 2004.